REMARKS/ARGUMENTS

This is a Response to the Office Action mailed May 22, 2003, in which a three (3) month Shortened Statutory Period for Response has been set, due to expire August 22, 2003. Enclosed is our check to cover the fee for a three-month extension of time, to November 22, 2003 (Saturday). Thirty (30) claims, including seven (7) independent claims, were paid for in the application. Claims 25-30 have been canceled. No claims have been amended or added. No new matter has been added to the application. No fee for additional claims is due by way of this Amendment. The Commissioner is authorized to charge any additional fees due by way of this Amendment, or credit any overpayment, to our Deposit Account No. 19-1090. Claims 1-24 are pending.

Election/Restrictions

During a telephone conversation with Examiner Nguyen on May 8, 2003, Applicants elected without traverse to prosecute the invention of Group I, claims 1-24.

In view of the above election, Applicants hereby cancel claims 25-30 without prejudice to the filing of any divisional, continuation, or continuation-in-part application.

Rejections Under 35 U.S.C. § 103

Claims 1-24 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,878,213 issued to Bittinger et al. (hereinafter Bittinger).

The present application is generally related to taking advantage of a distributed computing environments by distributing applications between client and server computers. The client computers include browser user interface programs that include user interfaces that communicate over networks with server applications running on the server computers. Some of these server applications would be enhanced by or necessitate duplex communication between the browser user interface and the server applications where simultaneous two-way communication occurs in both directions between the client and server computers.

Bittinger is directed to a time coherent caching system that attempts to reduce browser requests, servers responses and hence network traffic by caching information, and supplying the cached information in response to requests, if the cached information is not stale.

In particular, Bittinger discloses a first (client) computer 5 communicatively coupled to a second (server) computer 6 via an external communications link 35. Bittinger, Figure 11. The client computer runs a Web browser and the server computer runs a Web server. Bittinger, Figures 11 and 12. The client computer includes a client side intercept (CSI) module 30, while the server computer includes a server side intercept (SSI) module 40.

The CSI is coupled to the Web browser by a first pair of "real" sockets 65A, 65B, and is capable of creating additional pairs of real sockets 60A-64A, 60B-64B in response to requests from the Web browser. Bittinger, col. 19, lines 13-37. Bittinger, Figure 11. The SSI module is coupled to the Web server by pairs of real sockets 60C-64C, 60D-64D. Real multiplex sockets 36A, 36B are created in response to a request being sent by the client computer to the server computer over real sockets 37A, 37B, providing access to the external communications link. Bittinger, col. 19, lines 38-57.

Figure 12 illustrated the system in further detail. In particular, the CSI module includes a client side real socket manager 68, that creates simplex virtual sockets 70-74, applications of the CSI function 80-84 in response to the connection requests received at the first real socket 65b. Bittinger, col. 20, lines 36-63. The CSI manager creates multiplex virtual sockets 90-94 in response to data being received at the respective simplex virtual socket from the Web browser. Bittinger, col. 21, lines 29-44. Bittinger defines the "simplex socket" or "simplex virtual socket" as a socket which connected directly to either a single socket or application, and "multiplex socket" as a socket which connects to a plurality of other sockets, and thus carries out a multiplexing or demultiplexing function. Bittinger, col. 20, line 64–col. 21, line 3.

Similarly, the SSI module includes a server side real socket manager 69. Bittinger, col. 21, line 66-col. 22, line 13. The server side real socket manager creates applications of the SSI functions 85-89 and multiplex virtual sockets 95-99 if data is received from the Web browser. Bittinger, col. 22, lines 21-28. The SSI applications create simplex virtual sockets 75-79 after processing data. Bittinger, col. 22, lines 40-44.

After the response is received by the Web browser, the respective simplex virtual sockets 70-74, 95-99 and multiplex virtual sockets 90-94, 75-79 are closed, and respective applications 75-84 terminated. Bittinger, col. 24, lines 28-et seq.

Claim 1 recites, *inter alia*, "the client component and the server component configured such that each of the one or more instances of the client component is associated with one of the one or more instances of the server component to form a session for each association, each session having a session identifier and one or more sub-sessions designated as one or more data pipes, each data pipe being a sub-session of a particular session, having a pipe identifier, and configured to provide two independent data paths of *duplex data traffic* between the browser application communicatively linked to the instance of the client component associated with the particular session and the server application communicatively linked to the instance of the server component associated with the particular session." (Emphasis added.)

Claim 8 recites, *inter alia*, "the client component and the server component configured such that the instance of the client component is associated with the instance of the server component in an association to form a session, the session having a session identifier and a sub-session designated as a data pipe, the data pipe having a pipe identifier and configured to provide two independent data paths of *duplex data traffic* between the utility application communicatively linked to the instance of the client component and the server application communicatively linked to the instance of the server component." (Emphasis added.)

Claim 15 recites, *inter alia*, "a client component configured to run as one or more instances on the client computer, each instance of the client component being communicatively linked to one of the browser applications, each instance of the client component configured to be associated with an instance of the server component to form a session with a session identifier, the client component further configured to be associated with one or more data pipes, each data pipe being a sub-session of one of the sessions formed between instances of the client component and instances of the server component, each data pipe having a pipe identifier, each data pipe configured to provide two independent data paths of *duplex data traffic* between the browser application communicatively linked to the instance of the client component associated with the session of the data pipe and the server application communicatively linked to the instance of the server component associated with the session of the data pipe." (Emphasis added.)

-12-

Claim 20 recites, *inter alia*, "a server component configured to run as one or more instances on the server computer, each instance of the server component being communicatively linked to one of the server applications, each instance of the server component configured to be associated with an instance of the client component to form a session with a session identifier, the server component further configured to be associated with one or more data pipes, each data pipe being a sub-session of the session, each data pipe having a pipe identifier, each data pipe configured to provide two independent data paths of *duplex data traffic* between the browser application communicatively linked to the instance of the client component associated with the session of the data pipe and the server application communicatively linked to the instance of the server component associated with the session of the data pipe." (Emphasis added.)

In rejecting the claims, the Examiner has not drawn a clear correspondence between the various elements of the claims and the structures taught by Bittinger. For example, the Examiner relies on the browser 10 of Bittinger as teaching both the recited browser program and the recited one or more browser applications. Based on the relied on passages, the Examiner appears to draw a one-to-one correspondence between the CSI module and cache taught by Bittinger and the claimed client component. Likewise, the Examiner appears to draw a one-to-one correspondence between the SSI module and cache taught by Bittinger and the claimed server component. An alternative interpretation would draw a correspondence between the client side applications and the claimed client components, and between the server side applications and the claimed server components. Applicants respectfully request that the Examiner make clear which element in Bittinger he considers as corresponding to the claimed client and server components.

It appears that the Examiner draws a correspondence between the sockets taught by Bittinger and the claimed pipes. See Office Action, page 5, last line, "Socket function as pipe." While any given communications link between the Web browser and Web server (*e.g.*, Web browser 10↔real socket 60a↔real socket 60b↔simplex virtual socket 70↔client side application 80↔multiplex virtual socket 90↔real socket 36a↔real socket 36b↔multiplex virtual socket 95↔server side application 85↔simplex virtual socket 75↔real socket 60c↔real socket 60D↔Web server 20) taught by Bittinger appears capable of bi-directional

communications, Bittinger does not suggest or disclose each communications link being configured to provide *duplex* data traffic between the browser and server applications.

Not only is Bittinger silent with respect to duplex communications, but various passages of Bittinger strongly suggest that duplex communications would not be possible. For example, Bittinger must "flush" the data (step 335) queued in the multiple virtual socket after sending the request to the SSI, and "flush" the data (step 347) queued in the simplex virtual socket after sending the response to the browser. Bittinger, col. 21, lines 51-59, col. 24, lines 4-9, Figures 14, 17-1, and 17-2. Likewise, Bittinger must "flush" the data queued in the simplex virtual socket (step 365) after sending the request to the server, and must "flush" the data queued in the multiplex virtual socket (step 377) after sending the response to the CSI. Bittinger, col. 22, lines 49-55, col. 23, lines 22-29, Figures 15, 17-1, and 17-2. These flushes would likely not be necessary if there were two independent paths of duplex data traffic for each sub-session.

With respect to claims 4 and 12, Bittinger does not appear to disclose or suggest messages that contain both a pipe identifier identifying the data pipe and a pipe sequence number identifying an order of the messages in the duplex data traffic associated with the data pipe.

Conclusion

Overall, the cited reference does not singly, or in any motivated combination, teach or suggest the claimed features of the embodiments recited in independent claims 1, 8, 15, and 20, and thus such claims are allowable. Because the remaining claims depend from allowable independent claims 1, 8, 15, and 20, and also because they include additional limitations, such claims are likewise allowable. If the undersigned attorney has overlooked a relevant teaching in any of the references, the Examiner is requested to point out specifically where such teaching may be found.

In light of the above amendments and remarks, Applicants respectfully submit that all pending claims are allowable. Applicants, therefore, respectfully request that the Examiner reconsider this application and timely allow all pending claims. Examiner Nguyen is encouraged to contact Mr. Abramonte by telephone to discuss the above and any other distinctions between the claims and the applied references, if desired. If the Examiner notes any

informalities in the claims, he is encouraged to contact Mr. Abramonte by telephone to expediently correct such informalities.

Respectfully submitted,

Seed Intellectual Property Law Group PLLC

Frank Abramonte
Registration No. 38,066

FXA:asl

Enclosure:
    Postcard

701 Fifth Avenue, Suite 6300
Seattle, Washington 98104-7092
(206) 622-4900
Fax: (206) 682-6031

C:\NrPortbl\iManage\ANGIEL\434994_1.DOC